Amendments to the Specification:

Please replace paragraph **[0004]** with the following amended paragraph:

[0004] Switch fabrics and packet processors in high-performance broadband switches, such as Gigabit Ethernet switches or line cards, typically run at a fraction of their rated or maximum capacity. That is, typical processing loads do not require the full capacity of the switch fabrics and packet processors. Thus, it would be desirable to ~~provided~~ provide a scheme to allow such switch fabrics or packet processors to "oversubscribe" data to achieve more efficient usage of the processing capacity, where oversubscription means that the capacity of the data feed is larger than the capacity of data processing or switching.

Please replace paragraph **[0013]** with the following amended paragraph:

[0013] As shown in FIG. 1, the circuit **10** includes a plurality of ingress data ports **16 (16a, 16b)**, an aggregation module **18** coupled to the plurality of ingress data ports **16**, a memory **20** coupled to the aggregation module **18**, and an output data port **22** coupled to the aggregation module **18**. The aggregation module **18** may be implemented by a field programmable logic device (FPLD), field programmable gate array (FPGA), or the like. Each of the ingress data port **16 (16a** or **16b)** receives an input data stream **24 (24a** or **24b)** from a corresponding first processor **12 (12a** or **12b)**. Each of the input data streams **24 (24a, 24b)** is formed of ingress data packets. The aggregation module **18** is adapted to analyze and combine the plurality of input data ~~steams~~ streams **24 (24a, 24b)** into one aggregated data stream **26** in response to priority factors of the ingress data packets. The memory **20** is adapted to ~~stores~~ store analyzed data packets. The memory **20** may be an external buffer memory. The aggregated data stream **26** is output from the output data port **22** to the second processor **14**. Although FIG. 1 show two first processors **12**, the number of the first processors and the corresponding data streams is not limited to two.

Please replace paragraph **[0015]** with the following amended paragraph:

[0015] In accordance with one embodiment of the present invention, each of the first processors **12** and second processors **14** includes a logical interface providing logical interconnection between a Media Access Control sublayer (MAC) and a Physical layer (PHY), such as the 10 Gigabit Media Independent Interface (XGMII), through which data streams are received and transmitted. For example, the first processors **12** may be Layer-2 switching processors implementing Ethernet Maida Access Controllers and supporting the GMII, and the second processor **14** may be a data packet processor processing the aggregated packet data stream in the GMII format. Typically, the first processors **12** receive a receive (Rx) signal as the input data stream from transceivers, and the data flow from the first processors **12** to the second processor **14** through the aggregation module **18** forms a receive data path in the system. On the other hand, the data flow ~~form~~ from the second processor **14** to the first processors **12** typically forms a transmit (Tx) data path.

Please replace paragraph **[0019]** with the following amended paragraph:

[0019] The queue module **42** includes a plurality of priority queues **48** and selection logic **50**. Each of the priority queues **48** is provided for the corresponding priority class, and the selection logic **50** implements a queue scheme. For example, four (4) priority queue may be provided. The first and second packet analyzers **40a** and **40b** analyze and classify each of the ingress data packets into one of the priority classes based on the priority factors, and also generate a packet descriptor for each of the analyzed ingress data packets. The analyzed data packet is stored in the memory **20**. The packet descriptor contains a reference to a memory location of its analyzed data packet. The packet descriptor is placed in a priority queue **48** corresponding to the priority class of the data packet. The selection logic **50** arbitrates and ~~select~~ selects a packet descriptor from among the priority queues **48** in accordance with the queue scheme. Such a queue scheme includes strict fair queuing, weighted fair queuing, and the like.

Please replace paragraph **[0025]** with the following amended paragraph:

[0025] Similarly to the circuit **10** in FIGS. 1 and 2, the circuit **100** includes a plurality of ingress data ports **116 (116a, 116b)**, an aggregation module **118** coupled to the plurality of ingress data ports **116**, a memory **120** coupled to the aggregation module **118**, and an output data port **122** coupled to the aggregation module **118**. Each of the ingress data ports **116** receives an input data stream **124 (124a** or **124b)** from a corresponding first processor (not shown). Each of the input data streams **124 (124a, 124b)** is formed of ingress data packets, and each of the ingress data packets includes priority factors coded therein. The aggregation module **118** is adapted to analyze and combine the plurality of input data ~~steams~~ streams **124 (124a, 124b)** into one aggregated data stream **126** in response to the priority factors. The memory **120** is adapted to ~~stores~~ store analyzed data packets. The memory **120** may be an external buffer memory. The aggregated data stream **126** is output from the output data port **122** to the second processor (not shown). Although the number of the input data streams is not limited to two, the following description uses an example where two input data streams **124** are aggregated into one data stream **126**.

Please replace paragraph **[0028]** with the following amended paragraph:

[0028] As shown in FIG. 4, the aggregation module **118** includes a first packet analyzer **140a**, a second packet analyzer **140b**, a queue module **142**, a memory interface **144** including a first memory interface **144a** and a second memory interface **144b**, and an output module **146**. The first packet analyzer **140a** is coupled to the first data port **116a**, the first memory interface **144a**, and the queue module **142**. Similarly, the second packet analyzer **140b** is coupled to the second data port **116b**, the second memory interface **144b**, and the queue module **142**. The first and second packet analyzers **140a** and **140b** analyze and classify each of the ingress data packets into one of the priority classes based on the priority factors contained in the ingress data packet. The first and second packet analyzers **140a** and **140b** also generate a packet descriptor for each of the analyzed ingress data packets. The analyzed data packets are ~~store~~ stored in the memory **120**.

Please replace paragraph **[0029]** with the following amended paragraph:

[0029] As shown in FIG. 4, the external memory **120** may include a first memory unit (memory bank) **120a** and a second memory unit (memory bank) **120b** for the first input data stream **124a** and the second input data stream **124b**, respectively. In addition, the memory interface ~~140~~ **144** may also include a first memory interface ~~140a~~ **144a** for the first input data stream **124a** and a second memory interface ~~140b~~ **144b** for the second input data stream **124b**. Each of the memory unit may include a set of quad data rate (QDR) random access memories (RAMs) as shown in FIG. 4. It should be noted that write ports for the memory units **120a** and **120b** may be provided separately for the first and second input data streams **124a** and **124b**, and a read port may be common to both the first and second input data streams **124a** and **124b**.

Please replace paragraph **[0030]** with the following amended paragraph:

[0030] The packet descriptor contains a reference to a memory location of its analyzed data packet in the memory **120**. The packet descriptor is placed in the queue module **142**. The queue module **142** includes a plurality of priority queues **148** and selection logic **150**. Each of the priority queues **148** is provided for the corresponding priority class, and the packet descriptor is placed in the priority queue **148** corresponding to the priority class of its data packet. That is, packet descriptors of the ingress data packets for both of the first and second input data streams **124a** and **124b** are placed in the same priority queue **148** if they belong to the same priority class. The selection logic **150** implements a queue scheme, and arbitrates and ~~select~~ selects a packet descriptor from among the priority queues **148** in accordance with the queue scheme. Such a queue scheme includes strict fair queuing, weighted fair queuing, and the like.

Please replace paragraph **[0034]** with the following amended paragraph:

[0034] In accordance with one embodiment of the present invention, the priority of a data packet is assigned using per-port priority, VLAN priority, and protocol filter. For example, assume that

the ingress data packets are to be classified into four priority classes. Each priority factor of an ingress data packet may be assigned with a certain number such as 3, 2, 1, or 0, indicating the priority class, with number 3 indicating the highest priority. For example, each port number may be mapped onto one of the priority numbers. If the ingress data packet has been formatted with another priority queue scheme, such an external priority number, for example, a predefined VLAN priority number, may also be mapped onto one of the (internal) priority numbers 3, 2, 1, and 0. If the ingress data packet is a protocol packet, the priority factor associated with the protocol filter may be assigned with number 3. Then, the priority numbers assigned to respective factors of the data packet are "merged" or compared to each other and the highest priority number is determined as the ultimate priority number for that data packet. The data packet is classified according to the ultimate priority number. For example, if the ingress data packet is a protocol packet, it would be classified into the highest priority class even if other priority factors receives lower priority number.

Please replace paragraph **[0044]** with the following amended paragraph:

[0044] FIG. 9 schematically illustrates a method for aggregating a plurality of input data streams from first processors into one data stream for a second processor, in accordance with one embodiment of the present invention. The first processors and the second processor are provided on an electronic circuit substrate. A field programmable logic device (FPLD) coupled between the first processors and the second processor is provided **(350)**. An ingress data interface is provided between each of the first processors and the FPLD **(352)**. Each ingress data interface is adapted to couple an input data stream from a corresponding first processor to the FPLD. For example, the ingress data interface may be the XGMII supported by the first processor. Each input data stream is formed of ingress data packets, and each ingress data packet includes priority factors coded therein, as described above. An output data interface is also provided between the FPLD and the second processor **(354)**, which is adapted to couple the aggregated data stream to the second processor. For example, the output data interface may be a XGMII supported by the second processor. A memory coupled to the FPLD is also provided **(356)**, which is adapted to

store analyzed data packets. The FPLD is programmed such that the FPLD analyzes and combines the plurality of input data ~~steams~~ streams into one aggregated data stream in response to the priority factors **(360)**. The programmed FPLD performs the aggregation function for the Rx data ~~steam~~ stream as described above in detail with respect to other embodiments. The FPLD may also be programmed such that it also performs forwarding functions for the Tx data stream as described above, with providing an input data interface for receiving the Tx data from the second processor, and output interfaces for outputting output data streams to the first processors.